IN THE CLAIMS:

Set forth below in ascending order, with status identifiers, is a complete listing of all claims currently under examination. Changes to any amended claims are indicated by strikethrough and underlining. This listing also reflects any cancellation and/or addition of claims.

1. (currently amended) In a memory system, a method for maintaining multiple copies of data, the method comprising

detecting a write by a first processor to a memory associated with a second processor;

determining whether the write to said memory is directed to a memory space to be shadowed; [[and]]

when the write to said memory is detected to a memory space to be shadowed, identifying data for the write as data used by multiple processors including said second processor; and

cloning the write and storing the cloned write in a second memory.

2. (original) The method of claim 1 wherein said memory is a local memory and said second memory is a main memory.

3. (currently amended) The method of claim 1 wherein said storing comprises translating an address of the shadowed region of the ~~local~~ memory to an address in the ~~system~~ main memory.

4. (currently amended) A method for enhancing memory access in a multiple processing unit system including a central processing unit, a second processor, a local memory associated with the second processor and a system memory, the method comprising:

associating a shadow mask bit and a region of the local memory;

identifying the region of the local memory as a region for shadowing by placing the associated shadow mask bit in a first state;

designating data written into said region as data being used by multiple processors including said second processor;

detecting a write operation to the region of local memory by the central processing unit;

detecting that the shadow mask bit associated with the region is in said first state;

cloning the write operation; and

writing the cloned write operation to system memory.

5. (original) The method of claim 4 wherein the writing the cloned write operation comprises

translating an address of the first region in local memory to an address in system memory; and

storing data from the write operation to the address resulting from the translating operation.

6. (original) The method of claim 4 further comprising:

associating a second region of the local memory with a second shadow mask bit;

identifying the second region of the local memory as a region that is not to be shadowed by placing the associated second shadow mask bit in a second state;

detecting a write operation to the second region of local memory by the central processing unit;

detecting that the second shadow mask bit is in the second state; and

processing the write operation to the local memory without creating a shadow copy of the write operation in the system memory.

7. (original) The method of claim 6 further comprising:

resetting the second shadow mask bit to the first state;

detecting a second write operation to the second region of local memory;

detecting that the second shadow mask bit is in the first state;

cloning the second write operation to the second region of local memory; and

writing the cloned second write operation to system memory.

8. (currently amended) In a graphics processing system, including a graphics processing unit, a local memory associated with the graphics processing unit, a co-processor and a system memory, a method for memory management, the method comprising:

initiating an operation to write vertex data in the local memory;

detecting that the region of local memory to which the vertex data is to be written is a region to be shadowed, said detecting comprising:

identifying data for the write as data being used by the co-processor and the graphics processing unit;

cloning the vertex data to be written to local memory; and

routing the cloned vertex data to system memory.

9. (original) The method of claim 8 wherein said detecting comprises determining a shadow mask bit associated with the region of local memory to be in a state indicative of requiring a shadow operation.

10. (original) The method of claim 9 wherein said routing includes,

translating an address of the region in local memory into an address in system memory; and

writing the vertex data to the system memory at the address provided by the translating operation.

11. (original) The method of claim 8 wherein said routing includes,

translating an address of the region in local memory into an address in system memory; and

writing the vertex data to the system memory at the address provided by the translating operation.

12. (currently amended) A memory system comprising:

a central processing unit;

a second processor;

a local memory associated with said second processor;

a shadow unit coupled to said central processing unit and including a shadow bit mask having a plurality of bits, each bit associated with a region of the local memory designated to store data used by multiple processors including said second processor; and

an address translator coupled to said shadow unit.

13. (original) The system of claim 12 wherein said address translator includes a translation cache having a pool of translations.

14. (currently amended) A graphics processing system comprising:

a central processing unit;

a graphics processing unit;

a graphics processing unit frame buffer;

a coprocessor unit;

a coprocessing unit frame buffer; and

a shadow unit coupled to said central processing unit and coprocessing unit frame buffer, said shadow unit including a shadow mask adapted to track regions in said graphics processing unit frame buffer to be shadowed in the coprocessing unit frame buffer, said regions being designated to store data used by both said coprocessor unit and said graphics processing unit.

15. (new) The method of claim 4 wherein the writing the cloned write operation comprises:

performing a single translation of an address in local memory to a single translated address in system memory; and

storing data from the write operation in a translated region specified by said single translated address.

16. (new) The method of claim 1 wherein said multiple processors include a coprocessor.

17. (new) The method of claim 16 further comprising accessing said second memory by said coprocessor to use said data in said second memory rather than said first memory.

18. (new) The method of claim 16 further comprising translating a memory address in said coprocessor.

19. (new) The method of claim 4 wherein writing the cloned write operation includes diverting access by another of said multiple processors from said local memory to said system memory.

20. (new) The method of claim 19 wherein diverting access of said another of said multiple processors includes reducing data traffic between said local memory and said second processor, thereby enhancing system performance.